DETAILED ACTION

1. The amendment filed on 2/11/2008 has been entered and fully considered.

2. Claims 1-6 and 9-20 are pending. Claims 1, 14, and 15 are the base independent claims.

Claim Objections

3. Claims 1, 14, and 15 are objected to because of the following informalities: in line 12 of claim 1, in line 13 of claim 14, and in line 16 of claim 15 the phrase "plurality of processors a pointer to a storage location" needs to be replaced with the phrase ---- plurality of processors using a pointer to a storage location ----. Appropriate correction is required.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. **Claims 1-6, and 9-20** are rejected under 35 U.S.C. 103(a) as being unpatentable over Shanklin et al (US 6, 578, 147), in view of Salapura et al (US 6, 904, 040), Blair (US 6, 778, 495 B1) and Graham et al (US 7, 133, 405 B2).

Regarding **claim 1**, Shanklin'147 discloses a method for routing data packets for network flow analysis by a multi-processor system having a plurality of processors **(See Figure 2 and 3; Sensors 21 and 31 in Figures 2 and 3 respectively make up the**

multi-processor system), comprising: receiving a data packet, the data packet comprising data sufficient to identify a network connection with which the data packet is associated **(See Column 4:32-40 and Column 6:9-13)**; and assigning the data to one of the plurality of processors for analysis **(See Column 3:30, Column 5:22-29, 55-60 and Column 7:54-57).**

Shanklin'147 fails to disclose calculating a hash value based on the data sufficient to identify the network connection with which the data packet is associated and assigning the data based on the hash value to one of the plurality of processors for analysis by using a number of bits of the hash value, wherein the number of bits used is determined at least in part by the number of processors included in the plurality of processors.

However, the above mentioned claimed limitations are well known in the art as evidenced by Salapura'040. In particular, Salapura'040 discloses calculating a hash value based on the data sufficient to identify the network connection **(In Column 4:25-30 Salapura'040 discloses hash value calculation to identify connection)** with which the data packet is associated and assigning the data based on the hash value to one of the plurality of processors for analysis by using a number of bits of the hash value, wherein the number of bits used is determined at least in part by the number of processors **(See Columns 5:42-45, 6:18-21, 7:2-5 where Salapura'040 shows the packet being assigned to the processors based on hash value and the number of bits in the hash value corresponds to the number of processors involved.)** .

In view of the above, having the method of Shanklin'147 and then given the well established teaching of Salapura'040, it would have been obvious to one having ordinary skill in the art at the time of the invention was made to modify the method of Shanklin'147 as taught by Salapura'040, the motivation being hash value calculation simplifies address lookup as it is low cost to implement and saves processor time as stated in Salapura'040 in Column 1, Lines 40-60 and Column 2, Lines 1-2 and 50-53 and further distributing the workload among the processors on a per session basis allows it to outperform conventional network handlers in terms of cost and processing efficiency as further stated in Salapura'040 in Column 7, Lines 5-10.

Shanklin'147 fails to disclose the number of bits of the hash value used to identify the processors/links is not necessarily the total number of bits.

However, the above mentioned claimed limitations are well known in the art as evidenced by Blair'495. In particular, Blair'495 discloses the number of bits of the hash value used to identify the processors/links is not necessarily the total number of bits **(See Column 9, Lines 64-67 and Column 10, Lines 1-18)**.

In view of the above, having the method of Shanklin'147 and then given the well established teaching of Blair'495, it would have been obvious to one having ordinary skill in the art at the time of the invention was made to modify the method of Shanklin'147 as taught by Blair'495, the motivation for the modification to use only a portion of the hash value or result is that it allows the user to add links/processors (i.e. entities identified by the hash value) to the system without modifying the hashing function as stated by Blair in Column 10, Lines 1-5.

Shanklin'147 fails to disclose a method wherein the data packet is assigned to the one of the plurality of processors by storing in a work queue associated with the one of the plurality of processors, a pointer to a storage location in which data comprising the data packet is stored; and the processor is configured to read the pointer, use the pointer to read the data comprising the data packet directly from the storage location in which the data comprising the data packet is stored, use the data comprising the data packet to perform a network flow analysis with respect to a network flow with which the data packet is associated, and store in a return queue associated with the processor a data indicating that the processor is finished processing the data comprising the data packet; and wherein the data indicating that the processor is finished processing the data comprising the data packet is used to determine that the storage location is available to be used to store a subsequently received data comprising a subsequently received data packet.

Graham'405 discloses a method wherein the data packet is assigned to the one of the plurality of processors by storing in a work queue associated with the one of the plurality of processors **(Graham'405 shows in Figure 4 shows a receive work queues 400 and send work queues 402 associated a consumer associated with a processor 406. This is further shown in Figure 5 where each processor has a work queue.)**, a pointer to a storage location in which data comprising the data packet is stored; and the processor is configured to read the pointer **(See Column 6, Lines 18-25 and Column 8, Line 21)**, use the pointer to read the data comprising the data packet directly from the storage location in which the data comprising the data packet is

stored, use the data comprising the data packet to perform a network flow analysis with respect to a network flow with which the data packet is associated **(See Column 8, Lines 1-10 as Graham'405 details every queue uses pointers to read and write into a queue and manage a queue)**, and store in a return queue **(Graham'405 refers to return queues as completion queues and shows it as element 404 in Figure 4)** associated with the processor a data indicating that the processor is finished processing the data comprising the data packet; and wherein the data indicating that the processor is finished processing the data comprising the data packet is used to determine that the storage location is available **(Column 8, Lines 55-67)** to be used to store a subsequently received data comprising a subsequently received data packet **(Graham'405 clearly shows that when the processor in question finishes processing the packet stored in the work queue an indication is returned to the completion queue indicating availability of space as indicated in the work queue as detailed in Column 7, Lines 57-67 and Column 8, Lines 8-25)**.

In view of the above, having the method of Shanklin'147 and then given the well established teaching of Graham'405, it would have been obvious to one having ordinary skill in the art at the time of the invention was made to modify the method of Shanklin'147 as taught by Graham'405, the motivation for the modification is to use a zero processor-copy data transfer for realizing high bandwidth and low-latency communication as stated by Graham'405 in Column 9, Lines 25-30.

Regarding **claim 2**, Shanklin'147 discloses a method wherein the data in the data packet is sufficient to identify the network connection with which the data packet is associated comprises address data. **(See Column 3, Lines 25-26)**

Regarding **claim 3**, Shanklin'147 discloses wherein the data sufficient to identify the network connection with which the data packet is associated comprises address data associated with a source computer that sent the data packet and address data associated with a destination computer to which the data packet is addressed**. (See Column 3, Lines 25-26, Column 4 Lines 12-15 and 25-30)**

Regarding **claim 4**, Shanklin'147 discloses wherein the data packet is sent using the TCP/IP suite of protocols and the data sufficient to identify the network connection with which the data packet is associated comprises an IP address and port number associated with the source computer that sent the data packet and an IP address and port number associated with the destination computer to which the data packet is addressed. **(See Column 3, Lines 25-26, Column 4 Lines 12-15 and 25-30. Shanklin'147 discloses the packets are sent using the TCP/IP protocol and the rest of the limitation is inherent to the protocol because every IP packet contains source and destination address)**

Regarding **claim 5**, Shanklin'147 teaches all aspects of the claimed invention as set forth in the rejection of claim 1 but fails to disclose a method further comprising storing the data packet in host memory associated with the multi-processor system.

Salapura'040 discloses a method further comprising storing the data packet in host memory associated with the multi-processor system. **(See Figure 2, elements 14**

and 25 and Column 4:6-20 and Salapura'040 clearly shows storing packets in memory associated with the multi-processor system)

In view of the above, having the method of Shanklin'147 and then given the well established teaching of Salapura'040, it would have been obvious to one having ordinary skill in the art at the time of the invention was made to modify the method of Shanklin'147 as taught by Salapura'040, since Salapura'040 clearly states in Column 4, Lines 35-37 that the motivation to use a host memory shared by all processors is to reduce cost of using different memory with different controllers for different processors and Salapura uses a single DMA controller to interface with the different processors to store and retrieve data from the Direct Memory Access that serves as the host memory.

Regarding **claim 6**, Shanklin'147 teaches all aspects of the claimed invention as set forth in the rejection of claim 5 but fails to disclose a method, further comprising sending an interrupt message to a driver, the interrupt message comprising data identifying the storage location in host memory in which the data packet is stored.

Salapura'040 discloses a method, further comprising sending an interrupt message to a driver, the interrupt message comprising data identifying the storage location in host memory in which the data packet is stored. **(See Columns 1 Line 32 and Column 6, Lines 22-29)**

In view of the above, having the method of Shanklin'147 and then given the well established teaching of Salapura'040, it would have been obvious to one having ordinary skill in the art at the time of the invention was made to modify the method of Shanklin'147 as taught by Salapura'040, since the motivation for using an interrupt

message is to awaken a processor for processing data, the end result being savings in processor time and simplification of address lookup as stated in Salapura'040 in Columns 1 Line 32 and Column 6, Lines 22-29.

Regarding **claim 9**, Shanklin'147 teaches all aspects of the claimed invention as set forth in the rejection of claim 1 but fails to disclose a method wherein the work queue is a circular queue.

Salapura'040 discloses a method wherein the work queue is a circular queue **(See Column 4:10)**.

In view of the above, having the method of Shanklin'147 and then given the well established teaching of Salapura'040, it would have been obvious to one having ordinary skill in the art at the time of the invention was made to modify the method of Shanklin'147 as taught by Salapura'040, since the motivation for the modification being simplification of address lookup, reduce processor time and provides a more efficient packet handling method in that it keeps packets sequences belonging to the same session intact by assigning the packets to a specific work queue belonging to a specific processor as stated in Salapura'040 Column 1, Lines 45-61 and Column 2, Lines 50-67.

Regarding **claim 10**, Shanklin'147 discloses a method further comprising associating the data packet with one or more other data packets associated with the same network connection with which the received data packet is associated to recreate a network flow associated with the network connection **(See Column 3: 43-46 where the initial network connection associated with the packet is kept through out the flow).**

Regarding **claim 11**, Shanklin'147 discloses a method further comprising analyzing the network flow to determine if any security-related event has occurred. **(See Column 3, Lines 55-65 and Column 5, Lines 30-40)**

Regarding **claim 12**, Shanklin'147 discloses a method, wherein a security-related event is determined to have occurred if the network flow matches a pattern associated with a known attack. **(See Column 5, Lines 30-40, Column 6, Lines 4-8, and Column 7, Lines 60-65)**

Regarding **claim 13**, Shanklin'147 discloses a method wherein a security-related event is determined to have occurred if the network flow deviates from normal and permissible behavior under the network protocol under which the data packet was sent **(See Column 5, lines 30-40, Column 6, lines 4-8, and Column 7, lines 60-65 where Shanklin'147 discusses the conditions for a security related event that deviates from normal behavior)**.

Regarding **claim 14**, Shanklin'147 discloses a computer program product for routing data packets for network flow analysis by a multi-processor system, the computer program product being embodied in a computer readable medium and comprising computer instructions **(See Figure 2 and 3; Sensors 21 and 31 in Figures 2 and 3 respectively make up the multi-processor system)**, for: receiving a data packet, the data packet comprising data sufficient to identify a network connection with which the data packet is associated **(See Column 4:32-40 and Column 6:9-13)**; and assigning the data to one of the plurality of processors for analysis **(See Column 3:30, Column 5:22-29, 55-60 and Column 7:54-57).**

Shanklin'147 fails to disclose calculating a hash value based on the data sufficient to identify the network connection with which the data packet is associated and assigning the data based on the hash value to one of the plurality of processors for analysis by using a number of bits of the hash value, wherein the number of bits used is determined at least in part by the number of processors included in the plurality of processors.

However, the above mentioned claimed limitations are well known in the art as evidenced by Salapura'040. In particular, Salapura'040 discloses calculating a hash value based on the data sufficient to identify the network connection **(In Column 4:25-30 Salapura'040 discloses hash value calculation to identify connection)** with which the data packet is associated and assigning the data based on the hash value to one of the plurality of processors for analysis by using a number of bits of the hash value, wherein the number of bits used is determined at least in part by the number of processors **(See Columns 5:42-45, 6:18-21, 7:2-5 where Salapura'040 shows the packet being assigned to the processors based on hash value and the number of bits in the hash value corresponds to the number of processors involved.)** .

In view of the above, having the computer program product of Shanklin'147 and then given the well established teaching of Salapura'040, it would have been obvious to one having ordinary skill in the art at the time of the invention was made to modify the computer program product of Shanklin'147 as taught by Salapura'040, the motivation being hash value calculation simplifies address lookup as it is low cost to implement and saves processor time as stated in Salapura'040 in Column 1, Lines 40-60 and Column

2, Lines 1-2 and 50-53 and further distributing the workload among the processors on a per session basis allows it to outperform conventional network handlers in terms of cost and processing efficiency as further stated in Salapura'040 in Column 7, Lines 5-10.

Shanklin'147 fails to disclose the number of bits of the hash value used to identify the processors/links is not necessarily the total number of bits.

However, the above mentioned claimed limitations are well known in the art as evidenced by Blair'495. In particular, Blair'495 discloses the number of bits of the hash value used to identify the processors/links is not necessarily the total number of bits **(See Column 9, Lines 64-67 and Column 10, Lines 1-18)**.

In view of the above, having the computer program product of Shanklin'147 and then given the well established teaching of Blair'495, it would have been obvious to one having ordinary skill in the art at the time of the invention was made to modify the computer program product of Shanklin'147 as taught by Blair'495, the motivation for the modification to use only a portion of the hash value or result is that it allows the user to add links/processors (i.e. entities identified by the hash value) to the system without modifying the hashing function as stated by Blair in Column 10, Lines 1-5.

Shanklin'147 fails to disclose a computer program product wherein the data packet is assigned to the one of the plurality of processors by storing in a work queue associated with the one of the plurality of processors, a pointer to a storage location in which data comprising the data packet is stored; and the processor is configured to read the pointer, use the pointer to read the data comprising the data packet directly from the storage location in which the data comprising the data packet is stored, use the data

comprising the data packet to perform a network flow analysis with respect to a network flow with which the data packet is associated, and store in a return queue associated with the processor a data indicating that the processor is finished processing the data comprising the data packet; and wherein the data indicating that the processor is finished processing the data comprising the data packet is used to determine that the storage location is available to be used to store a subsequently received data comprising a subsequently received data packet.

Graham'405 discloses a computer program product wherein the data packet is assigned to the one of the plurality of processors by storing in a work queue associated with the one of the plurality of processors **(Graham'405 shows in Figure 4 shows a receive work queues 400 and send work queues 402 associated a consumer associated with a processor 406. This is further shown in Figure 5 where each processor has a work queue.)**, a pointer to a storage location in which data comprising the data packet is stored; and the processor is configured to read the pointer **(See Column 6, Lines 18-25 and Column 8, Line 21)**, use the pointer to read the data comprising the data packet directly from the storage location in which the data comprising the data packet is stored, use the data comprising the data packet to perform a network flow analysis with respect to a network flow with which the data packet is associated **(See Column 8, Lines 1-10 as Graham'405 details every queue uses pointers to read and write into a queue and manage a queue)**, and store in a return queue **(Graham'405 refers to return queues as completion queues and shows it as element 404 in Figure 4)** associated with the processor a data indicating

that the processor is finished processing the data comprising the data packet; and wherein the data indicating that the processor is finished processing the data comprising the data packet is used to determine that the storage location is available **(Column 8, Lines 55-67)** to be used to store a subsequently received data comprising a subsequently received data packet **(Graham'405 clearly shows that when the processor in question finishes processing the packet stored in the work queue an indication is returned to the completion queue indicating availability of space as indicated in the work queue as detailed in Column 7, Lines 57-67 and Column 8, Lines 8-25)**.

In view of the above, having the computer program product of Shanklin'147 and then given the well established teaching of Graham'405, it would have been obvious to one having ordinary skill in the art at the time of the invention was made to modify the computer program product of Shanklin'147 as taught by Graham'405, the motivation for the modification is to use a zero processor-copy data transfer for realizing high bandwidth and low-latency communication as stated by Graham'405 in Column 9, Lines 25-30.

Regarding **claim 15**, Shanklin'147 discloses a system for routing data packets for network flow analysis by a multi-processor system having a plurality of processors **(See Figure 2 and 3; Sensors 21 and 31 in Figures 2 and 3 respectively make up the multi-processor system)**, comprising: receiving a data packet, the data packet comprising data sufficient to identify a network connection with which the data packet is associated **(See Column 4:32-40 and Column 6:9-13)**; and assigning the data to one

of the plurality of processors for analysis **(See Column 3:30, Column 5:22-29, 55-60 and Column 7:54-57).**

Shanklin'147 fails to disclose calculating a hash value based on the data sufficient to identify the network connection with which the data packet is associated and assigning the data based on the hash value to one of the plurality of processors for analysis by using a number of bits of the hash value, wherein the number of bits used is determined at least in part by the number of processors included in the plurality of processors.

However, the above mentioned claimed limitations are well known in the art as evidenced by Salapura'040. In particular, Salapura'040 discloses calculating a hash value based on the data sufficient to identify the network connection **(In Column 4:25-30 Salapura'040 discloses hash value calculation to identify connection)** with which the data packet is associated and assigning the data based on the hash value to one of the plurality of processors for analysis by using a number of bits of the hash value, wherein the number of bits used is determined at least in part by the number of processors **(See Columns 5:42-45, 6:18-21, 7:2-5 where Salapura'040 shows the packet being assigned to the processors based on hash value and the number of bits in the hash value corresponds to the number of processors involved.)** .

In view of the above, having the system of Shanklin'147 and then given the well established teaching of Salapura'040, it would have been obvious to one having ordinary skill in the art at the time of the invention was made to modify the system of Shanklin'147 as taught by Salapura'040, the motivation being hash value calculation

simplifies address lookup as it is low cost to implement and saves processor time as stated in Salapura'040 in Column 1, Lines 40-60 and Column 2, Lines 1-2 and 50-53 and further distributing the workload among the processors on a per session basis allows it to outperform conventional network handlers in terms of cost and processing efficiency as further stated in Salapura'040 in Column 7, Lines 5-10.

Shanklin'147 fails to disclose the number of bits of the hash value used to identify the processors/links is not necessarily the total number of bits.

However, the above mentioned claimed limitations are well known in the art as evidenced by Blair'495. In particular, Blair'495 discloses the number of bits of the hash value used to identify the processors/links is not necessarily the total number of bits **(See Column 9, Lines 64-67 and Column 10, Lines 1-18)**.

In view of the above, having the system of Shanklin'147 and then given the well established teaching of Blair'495, it would have been obvious to one having ordinary skill in the art at the time of the invention was made to modify the system of Shanklin'147 as taught by Blair'495, the motivation for the modification to use only a portion of the hash value or result is that it allows the user to add links/processors (i.e. entities identified by the hash value) to the system without modifying the hashing function as stated by Blair in Column 10, Lines 1-5.

Shanklin'147 fails to disclose a system wherein the data packet is assigned to the one of the plurality of processors by storing in a work queue associated with the one of the plurality of processors, a pointer to a storage location in which data comprising the data packet is stored; and the processor is configured to read the pointer, use the

pointer to read the data comprising the data packet directly from the storage location in which the data comprising the data packet is stored, use the data comprising the data packet to perform a network flow analysis with respect to a network flow with which the data packet is associated, and store in a return queue associated with the processor a data indicating that the processor is finished processing the data comprising the data packet; and wherein the data indicating that the processor is finished processing the data comprising the data packet is used to determine that the storage location is available to be used to store a subsequently received data comprising a subsequently received data packet.

Graham'405 discloses a system wherein the data packet is assigned to the one of the plurality of processors by storing in a work queue associated with the one of the plurality of processors **(Graham'405 shows in Figure 4 shows a receive work queues 400 and send work queues 402 associated a consumer associated with a processor 406. This is further shown in Figure 5 where each processor has a work queue.)**,

a pointer to a storage location in which data comprising the data packet is stored; and the processor is configured to read the pointer **(See Column 6, Lines 18-25 and Column 8, Line 21)**,

use the pointer to read the data comprising the data packet directly from the storage location in which the data comprising the data packet is stored, use the data comprising the data packet to perform a network flow analysis with respect to a network flow with which the data packet is associated **(See Column 8, Lines 1-10 as**

Graham'405 details every queue uses pointers to read and write into a queue and manage a queue), and

store in a return queue **(Graham'405 refers to return queues as completion queues and shows it as element 404 in Figure 4)** associated with the processor a data indicating that the processor is finished processing the data comprising the data packet; and wherein the data indicating that the processor is finished processing the data comprising the data packet is used to determine that the storage location is available **(Column 8, Lines 55-67)** to be used to store a subsequently received data comprising a subsequently received data packet **(Graham'405 clearly shows that when the processor in question finishes processing the packet stored in the work queue an indication is returned to the completion queue indicating availability of space as indicated in the work queue as detailed in Column 7, Lines 57-67 and Column 8, Lines 8-25)**.

In view of the above, having the system of Shanklin'147 and then given the well established teaching of Graham'405, it would have been obvious to one having ordinary skill in the art at the time of the invention was made to modify the system of Shanklin'147 as taught by Graham'405, the motivation for the modification is to use a zero processor-copy data transfer for realizing high bandwidth and low-latency communication as stated by Graham'405 in Column 9, Lines 25-30.

Regarding **claim 16**, Shanklin'147 discloses a system wherein the data sufficient to identify the network connection with which the data packet is associated comprises address data associated with a source computer that sent the data packet and address

data associated with a destination computer to which the data packet is addressed **(See Columns 3:23-25, 4:32-40, 6:9-13, and 7:20-27. It should be noted that all IP packets have headers and each header has a source and destination address).**

Regarding **claim 17**, it is noted that the limitations of claim 17 corresponds to that of claim 16 as discussed above, please see the Examiner's comments with respect to claim 16 as set forth in the rejection above.

Regarding **claim 18**, Shanklin'147 discloses a system, wherein the data packet is sent using the TCP/IP suite of protocols and the data sufficient to identify the network connection with which the data packet is associated comprises an IP address and port number associated with the source computer that sent the data packet and an IP address and port number associated with the destination computer to which the data packet is addressed. **(In Column 4, Lines 12-32 Shanklin'147 discloses that his system uses the TCP/IP suite of protocols including TCP, UDP, IP and ICMP. Examiner takes Official Notice that the TCP and UDP protocols provide port number associated with the source and the destination while IP protocol provides the IP address of the source as well as the destination. Please refer to Newton's Telecom dictionary 16th edition on pages 838-839 for further support)**

Regarding **claim 19,** Shanklin'147 discloses a system, wherein the driver is further configured to associate the data packet with one or more other data packets associated with the same network connection with which the received data packet is associated to recreate a network flow associated with network connection. **(See**

Column 7, Lines 54-59 Shanklin'147 disclose associating packets with specific network connections)

Regarding **claim 20**, Shanklin'147 discloses a system, wherein the driver is further configured to analyze the network flow to determine if any security-related event has occurred **(See Column 6, Lines 47-56).**

Response to Arguments

6. Applicant's arguments with respect to claims 1, 14, and 15 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

7. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to HABTE MERED whose telephone number is (571)272-6046. The examiner can normally be reached on Monday to Friday 9:30AM to 5:00PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Aung S. Moe can be reached on 571 272 7314. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Aung S. Moe/
Supervisory Patent Examiner, Art Unit 2616

Habte Mered
Examiner
Art Unit 2616
06-07-08